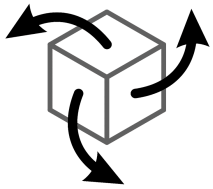
3-Space Sensor



3-Space Sensor HiPerGyro

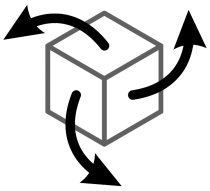
High Performance
Attitude & Heading Reference System

User's Manual

YEI Technology
630 Second Street
Portsmouth, Ohio 45662

www.YeiTechnology.com
www.3SpaceSensor.com

Patents Pending
©2007-2012 Yost Engineering, Inc.
Printed in USA



3-Space Sensor HiPerGyro

Miniature Attitude & Heading Reference System

User's Manual

YEI Technology
630 Second Street
Portsmouth, Ohio 45662

www.YeiTechnology.com
www.3SpaceSensor.com

Toll-Free: 888-395-9029
Phone: 740-355-9029

Patents Pending
©2007-2012 Yost Engineering, Inc.
Printed in USA

Table of Contents

1. Usage/Safety Considerations....1
    1.1 Usage Conditions....1
    1.2 Technical Support and Repairs....1
2. Overview of the YEI 3-Space Sensor HiPerGyro....2
    2.1 Introduction....2
    2.2 Applications....2
    2.3 Key Features....2
    2.4 Hardware Overview....3
    2.5 Block Diagram of Sensor Operation....4
    2.6 Specifications*....5
    2.7 Physical Dimensions....6
    2.8 Axis Assignment....6
3. Description of the 3-Space Sensor HiPerGyro....7
    3.1 Navigation Grade Gyroscope....7
        3.1.1 Fusion....7
    3.2 Orientation Estimation....7
        3.2.1 Component Sensors....7
        3.2.2 Scale, Bias, and Cross-Axis Effect....8
        3.2.3 Reference Vectors....8
        3.2.4 Reference Orientation/Taring....8
    3.3 Communication....8
    3.4 Sensor Settings....8
        3.4.1 Committing Settings....8
        3.4.2 Settings and Defaults....9
4. HiPerGyro Usage/Protocol....10
    4.1. Usage Overview....10
        4.1.1 Protocol Overview....10
        4.1.2 Computer Interfacing Overview....10
    4.2. Protocol Packet Format....11
        4.2.1 Binary Packet Format....11
        4.2.2 ASCII Text Packet Format....12
    4.3. Command Chart....13
        4.3.1 Filtered Data Commands....13
        4.3.2 Raw Data Commands....13
        4.3.3 Calibrated Data Commands....14
        4.3.4 Orientation Fusion Commands....14
        4.3.5 Gyro Fusion Commands....15
        4.3.6 Calibration Commands....16
        4.3.7 System Fusion Commands....17
Appendix....18
    USB Connector....18
    RS422 Connector....18
    Hex / Decimal Conversion Chart....18

1. Usage/Safety Considerations

1.1 Usage Conditions

- Do not use the 3-Space Sensor in any system on which people's lives depend(life support, weapons, etc.)
- Because of its reliance on a compass, the 3-Space Sensor's orientation estimate will not work properly near the earth's north or south pole.
- Because of its reliance on a compass and accelerometer, the 3-Space Sensor's orientation estimate will not work properly in outer space or on planets with no magnetic field.
- Care should be taken when using the 3-Space Sensor in a car or other moving vehicle, as the disturbances caused by the vehicle's acceleration may cause the sensor to give inaccurate readings.
- Because of its reliance on a compass, care should be taken when using the 3-Space Sensor near ferrous metal structures, magnetic fields, current carrying conductors, and should be kept about 6 inches away from any computer screens or towers.

1.2 Technical Support and Repairs

YEI provides technical and user support via our toll-free number (888-395-9029) and via email (support@YostEngineering.com). Support is provided for the lifetime of the equipment. Requests for repairs should be made through the Support department. For damage occurring outside of the warranty period or provisions, customers will be provided with cost estimates prior to repairs being performed.

2. Overview of the YEI 3-Space Sensor HiPerGyro

2.1 Introduction

The YEI 3-Space HiPerGyro is an integrated system that utilizes a novel arrangement of sensing elements combined with advanced algorithmic processing to produce an ultra-high-precision, high-reliability, miniature, navigation-grade Attitude and Heading Reference System (AHRS) / Inertial Measurement Unit (IMU). The system exhibits size, weight, and power characteristics similar to MEMS-based solutions yet achieves performance characteristics previously only attainable in laser-ring and fiber-optic systems.

The YEI 3-Space HiPerGyro utilizes a combination of gyroscope, accelerometer, magnetometer and barometer sensing elements in conjunction with advanced on-board processing and filtering algorithms to achieve ultra-accurate inertial measurements and attitude/heading outputs in real-time.

A proprietary multi-sensor-fusion approach combined with dynamic sensor-error and bias tracking allow for extreme accuracy and precision across a wide range of operating conditions.

The YEI 3-Space HiPerGyro unit features are accessible via a well-documented open communication protocol that allows access to all available sensor data and configuration parameters using either RS422 or USB 2.0 interfaces. Versatile commands allow access to raw sensor data, normalized sensor data, and filtered absolute and relative orientation outputs.

2.2 Applications

- Navigation
- Autonomous vehicles
- Antenna and platform stabilization
- Optical stabilization
- Personnel / pedestrian navigation and tracking
- Unmanned air/land/water vehicle guidance
- Marine motion sensing

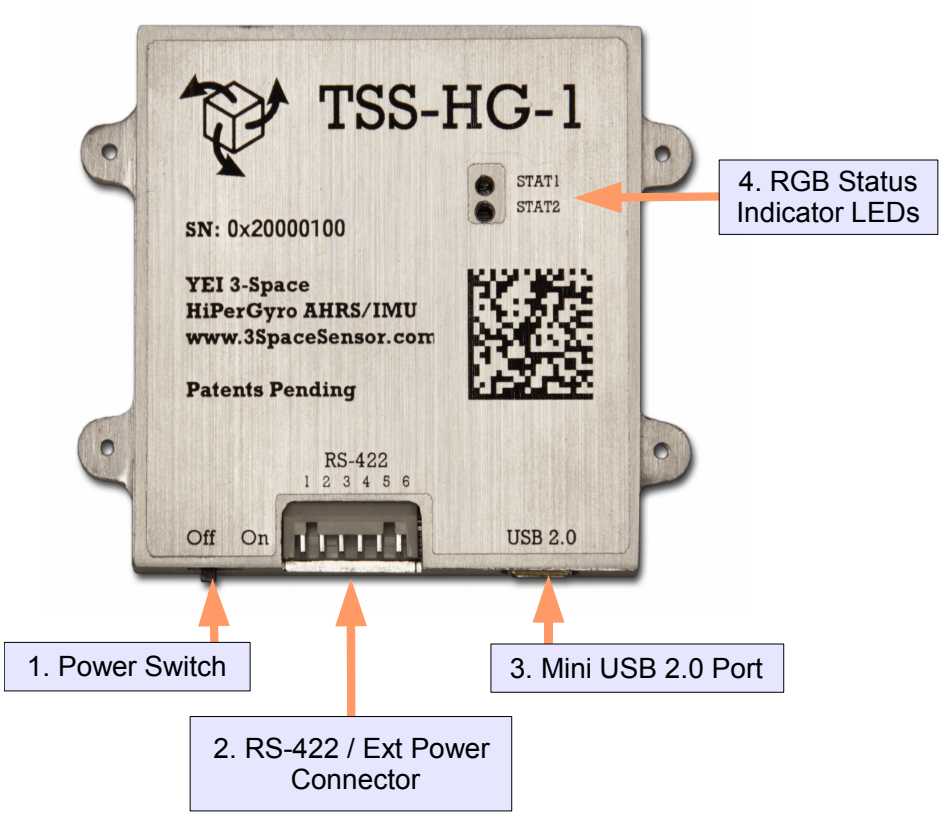
2.3 Key Features

The YEI 3-Space HiPerGyro has many features that allow it to be a flexible all-in-one solution for your high-accuracy inertial and orientation sensing needs. Below are some of the key features:

- Small, self-contained, ultra-high-performance AHRS at 54mm x 64mm x 10mm and <40 grams
- Optical gyro IMU performance characteristics at MEMS size, weight, and power.
- Fast sensor update and filter rate allow use in real-time applications, including navigation, stabilization, and guidance
- Advanced integrated on-board multi-sensor fusion provides ultra-high accuracy.
- Proprietary dynamic sensor-error tracking and bias tracking automatically minimize the effects of sensor noise and sensor error across a wide range of operating conditions.
- Proprietary bias-tracking achieves gyro bias drift of less than 0.001°/hr for all axes and gyro ARW of less than 0.00005°/√hr
- Robust open protocol allows commands to be sent in human readable form, or, more quickly, in machine readable form
- Orientation output format available in absolute or relative terms in multiple formats (quaternion, rotation matrix, axis angle, two-vector)

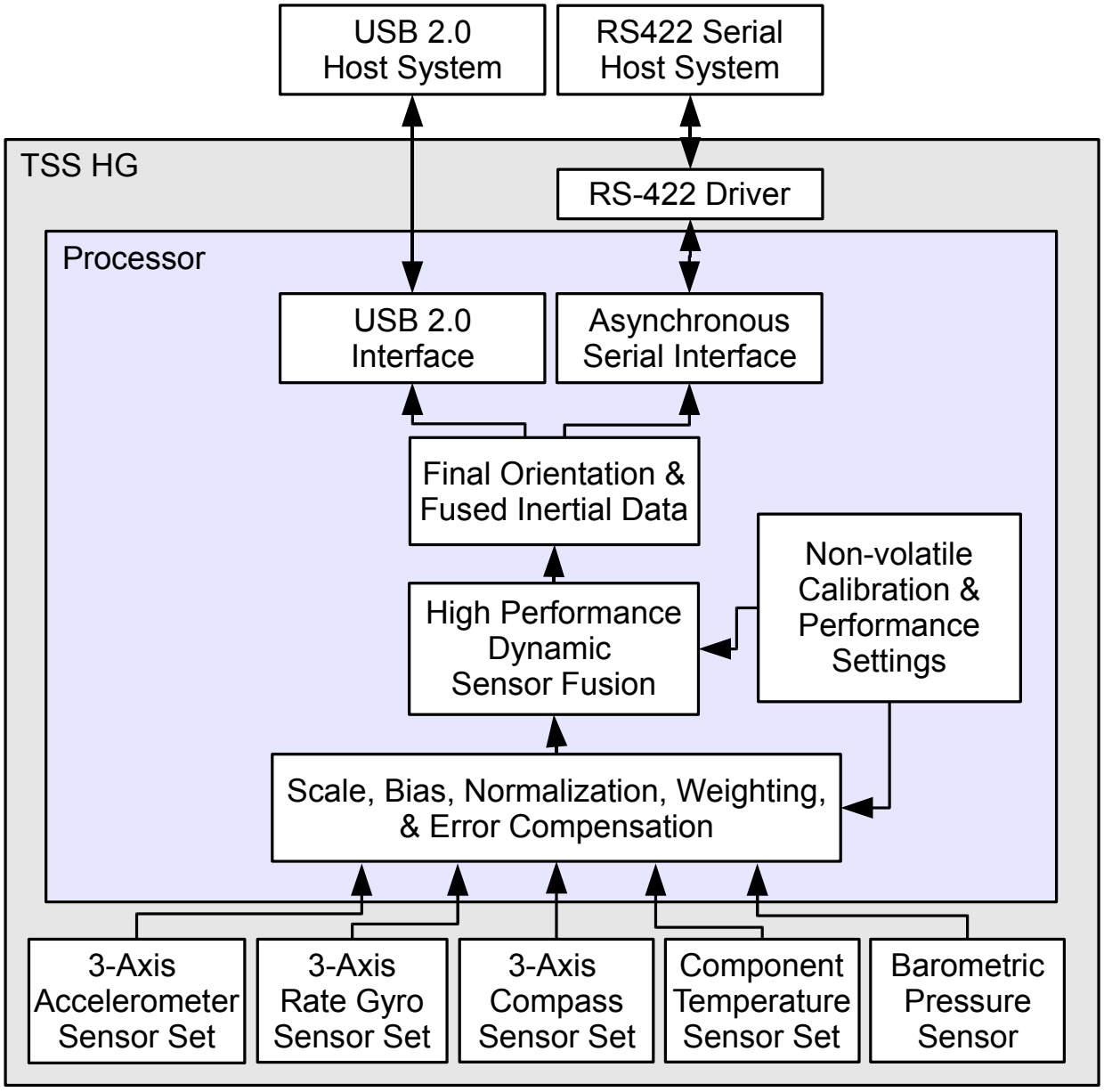
- Absolute or custom reference axes
- Includes barometric sensor to aid in altitude estimation.
- Flexible communication options: USB 2.0 or Full Duplex RS422
- Communication through a virtual COM port
- Upgradeable firmware
- RGB status LEDs

2.4 Hardware Overview



1. **Power Switch** – The 3-Space Sensor HiPerGyro can be switch on and off by using the power switch.
2. **RS-422 Connector -** The 3-Space Sensor HiPerGyro allows for external power and full-duplex RS-422 communication via connection to this port. The RS-422 connector provides for both power and communication signals.
3. **USB Connector** – The 3-Space Sensor HiPerGyro uses a 5-pin mini USB connector to connect to a computer via USB. The USB connector provides for both power and communication signals.
4. **Indicator LEDs** – The 3-Space Sensor HiPerGyro includes two RGB status LEDs that can provide visual status feedback during operation.

2.5 Block Diagram of Sensor Operation



2.6 Specifications*

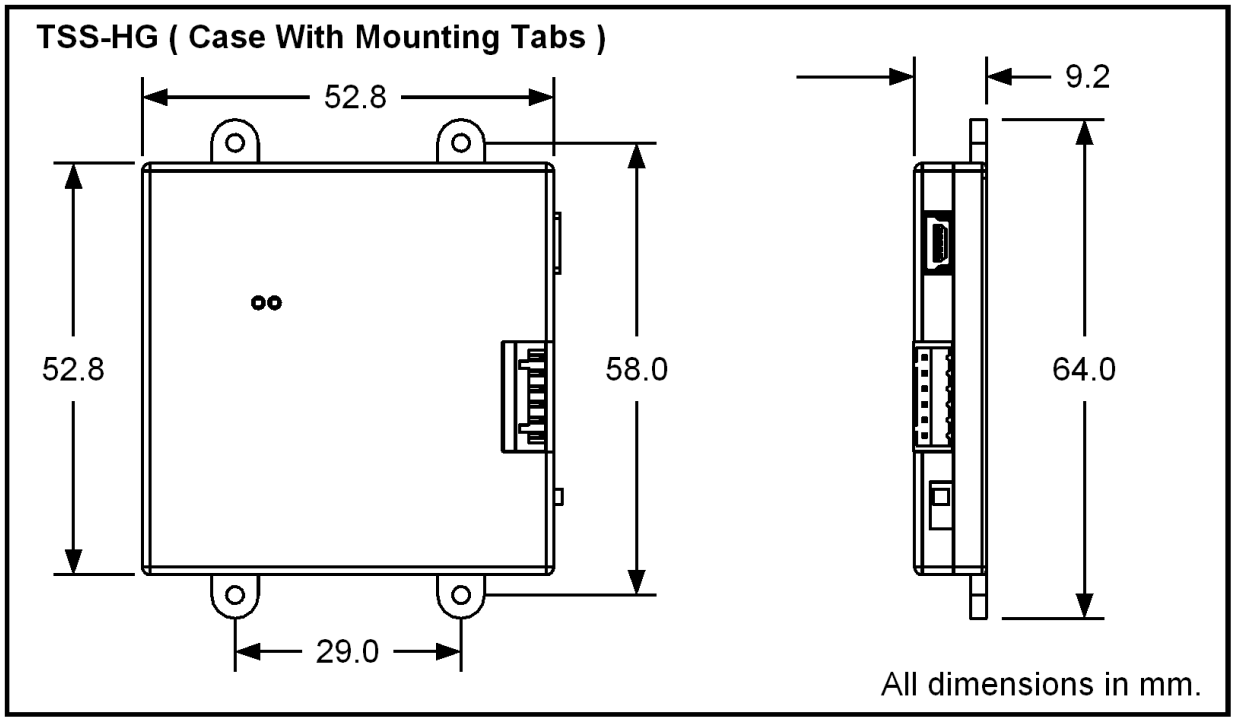
General	
Part number	TSS-HG-1
Dimensions	52.8mm x 64mm x 9.2mm (2.08in x 2.525in x 0.36in)
Weight	<45 grams (1.6 oz)
Supply voltage	+5v USB or +3.3v~+6v External
Power requirement	< 0.72W (<150mA @ 5VDC)
Communication interfaces	USB 2.0, RS422(Full Duplex)
Filter update rate[1]	180Hz with full performance filtering and bias tracking
Orientation output	absolute & relative quaternion, Euler angles, axis angle, rotation matrix, two vector
Other output	raw sensor data, normalized sensor data, calibrated sensor data, barometric pressure, temperature
Shock survivability	5000g
Temperature range	-40C ~ 85C (-40F ~ 185F)
Sensor[2]	
Orientation range	360º about all axes
Orientation accuracy	TBD
Orientation resolution	<0.02º
Orientation repeatability	0.021º for all orientations
Accelerometer scale	±2g/±4g/±8g selectable (other ranges up to ±400g available)
Accelerometer resolution	16-bit effective
Accelerometer noise density	24.8 µg/√ Hz
Accelerometer sensitivity	0.00024g/digit for ±2g range 0.00048g/digit for ±4g range 0.00096g/digit for ±8g range
Accelerometer temperature sensitivity	±0.008%/°C
Gyro scale	±250/±500/±2000 º/sec selectable
Gyro resolution	20-bit effective
Gyro angular random walk[3]	<0.00005º/√ hr
Gyro bias stability[3]	<0.001º/hr average for all axes
Gyro sensitivity	0.00875º/sec/digit for ±250º/sec 0.01750º/sec/digit for ±500º/sec 0.070º/sec/digit for ±2000º/sec
Gyro non-linearity	0.2% full-scale
Gyro temperature sensitivity	±0.016%/°C
Compass scale	±10 Ga
Compass resolution	20-bit effective
Compass sensitivity	1 mGa/digit
Compass non-linearity	0.1% full-scale
Barometer resolution	10-bit
Barometer range	50 kPa to 115 kPa (Absolute)
Barometer sensitivity	0.15 kPa / digit
Barometer accuracy	±1 kPa

1. Depends upon communication mode and filter mode.
2. All performance data are preliminary and subject to change
3. With full filtering and bias tracking enabled

Specifications are subject to change.

* Some parameters may be application and configuration dependent.

2.7 Physical Dimensions



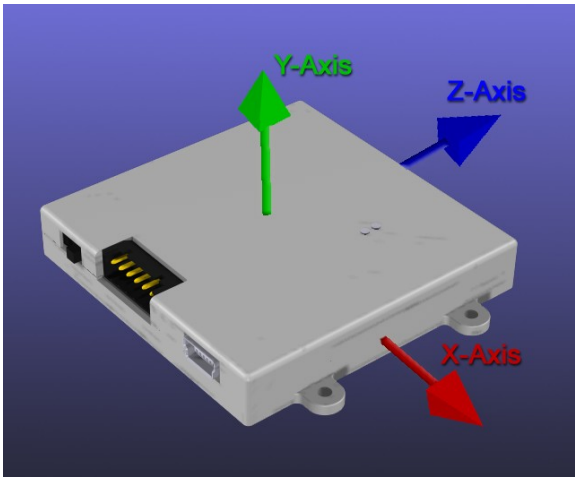
2.8 Axis Assignment

All YEI 3-Space Sensor product family members have re-mappable axis assignments and axis directions. This flexibility allows axis assignment and axis direction to match the desired end-use requirements.

The natural axes of the 3-Space Sensor HiPerGyro are as follows:

- The positive X-axis points out of the right hand side of the sensor, which is the side that is facing right when the LEDs face upward and USB plug faces towards you.
- The positive Y-axis points out of the top of the sensor, the side with the LEDs.
- The positive Z-axis points out of the front of the sensor, the side opposite the USB plug.

The natural axes are illustrated in the diagram below.



3. Description of the 3-Space Sensor HiPerGyro

3.1 Navigation Grade Gyroscope

The primary purpose of the HiPerGyro is to act as a navigation grade gyroscope. This means that it is highly accurate and does not suffer as much from error sources as other gyroscopes. In order to understand how to calibrate the HiPerGyro and use the output in a meaningful way, there are a few concepts about the sensor that should be understood. The following sections describe these concepts.

3.1.1 Fusion

The HiPerGyro uses an array of gyroscopes and other sensors in order to deliver a single high accuracy gyroscope reading. The process of combining these sensors is called fusion, and there are several options for each phase of fusion.

- **Fusion Method:** This determines how the outputs of the sensors are combined together. Currently, the options are **Averaged**, which will just take a simple average of the gyroscopes, or **Filtered**, where the outputs are put through a more complex algorithm which reduces the noise greatly, but is more computationally expensive than **Averaged**.
- **Deadband:** This sets up a deadband around the gyroscope readings so that, if it falls below a certain threshhold, the reading will be set to zero. This can be enabled or disabled.
- **Bias Tracking:** In order to reduce the effects of bias instability, the HiPerGyro can automatically track how the bias of the gyroscopes change over time, and how the bias of the final fusion reading changes as well. It also has a mode where gyroscopes will be compared to each other, allowing some bias tracking to occur while the sensor is in motion.
- **Orient Gyro Filter:** This will use the orientation estimate to further filter the gyroscope reading. This helps to reduce drift greatly, but makes it more susceptible to accelerations and magnetic effects. This can be enabled or disabled.

3.2 Orientation Estimation

Another purpose of the HiPerGyro is to estimate orientation. In order to understand how to handle this estimation and use it in a meaningful way, there are a few concepts about the sensor that should be understood. The following sections describe these concepts.

3.2.1 Component Sensors

The HiPerGyro estimates orientation by combining the data it gets from three types of sensors: a gyroscope, an accelerometer, and a compass. A few things you should know about each of these sensors:

- **Accelerometer:** This sensor measures the acceleration due to gravity, as well as any other accelerations that occur. Because of this, this sensor is at its best when the 3-Space Sensor is sitting still. Most jitter seen as the orientation of the sensor changes is due to shaking causing perturbations in the accelerometer readings. To account for this, by default, when the 3-Space Sensor is being moved, the gyroscope becomes more trusted(becomes a greater part of the orientation estimate), and the accelerometer becomes less trusted.
- **Gyroscope:** This sensor measures angular motion. It has no ability to give any absolute orientation information like the accelerometer or compass, and so is most useful for correcting the orientation during sensor motion. Its role during these times becomes vital, though, as the accelerometer readings can become unreliable during motion.
- **Compass:** This sensor measures magnetic direction. The readings from the compass and accelerometer are used together to form the absolute component of orientation, which is used to correct any short term changes the gyroscope makes. Its readings are much more stable than those of the accelerometer, but it can be adversely affected by any ferrous metal or magnetic objects. When the accelerometer is less trusted, the compass is treated in the same way so as to avoid updates to orientation based on partial absolute information.

3.2.2 Scale, Bias, and Cross-Axis Effect

The readings taken from each component sensor are not in a readily usable form. The compass and accelerometer readings are not unit vectors, and the gyroscope readings aren't yet in radians per second. To convert them to these forms, scale and bias must be taken into account. Scale is how much larger the range of data read from the component sensor is than the range of data should be when it is converted. For example, if the compass were to give readings in the range of -500 to 500 on the x axis, but we would like it to be in the range of -1 to 1, the scale would be 500. Bias is how far the center of the data readings is from 0. If another compass read from -200 to 900 on the x axis, the bias would be 350, and the scale would be 550. The last parameter used in turning this component sensor data into usable data is cross-axis effect. This is the tendency for a little bit of data on one axis of a sensor to get mixed up with the other two. This is an effect experienced by the accelerometer and compass. There are 6 numbers for each of these, one to indicate how much each axis is affected by each other axis. Values for these are generally in the range of 1 to 10%. These parameters are applied in the following order:

1) Bias is subtracted from each axis

2) The three axes are treated as a vector and multiplied by a matrix representing scale and cross-axis parameters

Factory calibration provides default values for these parameters for the accelerometer and compass, but end users may need to recalibrate the sensor to obtain parameters more appropriate to their location and application. The gyroscope is also factory calibrated, and these parameters should not change based on location or application.

3.2.3 Reference Vectors

In order to get an absolute estimation of orientation from the accelerometer and compass, the sensor needs a reference vector for each to compare to the data read from it. The default for these are the standard direction of gravity(down) and the standard direction of magnetic force(north), respectively.

The component sensor data and reference vectors are fed into a Kalman filter, which uses statistical techniques to optimally combine the data into a final orientation reading.

3.2.4 Reference Orientation/Taring

Given the results of the Kalman filter, the sensor can make a good estimation of orientation, but it will likely be offset from the actual orientation of the device by a constant angle until it has been given a reference orientation. This reference orientation tells the sensor where you would like its zero orientation to be. The sensor will always consider the zero orientation to be the orientation in which the plug is facing towards you and top(the side with LEDs on it) facing up. The sensor must be given a reference orientation that represents the orientation of the sensor when it is in the position in which you consider the plug to be towards you and the LEDs up. The act of giving this reference orientation to the sensor is called taring, just as some scales have a tare button which can be pressed to tell the scale that nothing is on it and it should read zero. For instructions on doing this, refer to command 64(0x40) to 66(0x42) on the command chart.

3.3 Communication

Obtaining data about orientation from the sensor or giving values for any of its settings is done through the sensor's communication protocol. The protocol can be used through either the USB port or the RS422 port. A complete description of how to use this protocol is given in section 4 of this document.

3.4 Sensor Settings

3.4.1 Committing Settings

Changes made to the HiPerGyro will not be saved unless they are committed. This allows you to make changes to the sensor and easily revert it to its previous state by resetting the chip. To commit your changes, call command 225(0xe1). Any changes relating to the multiple reference vector mode are an exception to this rule, as all these changes are saved immediately.

3.4.2 Settings and Defaults

Setting Name	Purpose	Default Value
RS232 Baud Rate	Determines the speed of RS232 communication	115200
RS422 Baud Rate	Determines the speed of RS422 communication	115200
LED 0 Color	Determines the color of the first LED	0,0,255
LED 1 Color	Determines the color of the second LED	0,255,0
Calibration Mode	Determine how sensors are calibrated	Vector Correction Mode(1)
Gyro Fusion Method	Determine how gyro fusion is carried out	Filtered(3)
Bias Tracking Mode	Determine how bias tracking is carried out	Both(3)
Orient Gyro Filter	Determine if orientation plays a role in gyroscope readings	Disabled(0)
Deadband	Determines if the gyroscope reading should have a deadband on it.	Disabled(0)

For more information on these settings, see the command chart.

4. HiPerGyro Usage/Protocol

4.1. Usage Overview

4.1.1 Protocol Overview

The 3-Space Sensor receives messages from the controlling system in the form of sequences of serial communication bytes called packets. For ease of use and flexibility of operation, two methods of encoding commands are provided: binary and text. Binary encoding is more compact, more efficient, and easier to access programmatically. ASCII text encoding is more verbose and less efficient yet is easier to read and easier to access via a traditional terminal interface. Both binary and ASCII text encoding methods share an identical command structure and support the entire 3-Space command set.

The 3-Space Sensor buffers the incoming command stream and will only take an action once the entire packet has been received and the checksum has been verified as correct(ASCII mode commands do not use checksums for convenience). Incomplete packets and packets with incorrect checksums will be ignored. This allows the controlling system to send command data at leisure without loss of functionality. The command buffer will, however, be cleared whenever the 3-Space Sensor is either reset or powered off/on.

Specific details of the 3-Space Sensor protocol and its control commands are discussed in the following pages.

4.1.2 Computer Interfacing Overview

When interfacing with a computer, the HiPerGyro presents itself as a COM port, which provides an interface by which the serial communication the protocol requires may happen. The name of this COM port is specific to the operating system being used, and the communication interface being used. It is possible to use multiple HiPerGyro on a single computer. Each will be assigned its own COM port. The easiest way to find out which COM port belongs to a certain sensor is to take note of what COM port appears when that sensor is plugged in(provided the drivers have been installed on that computer already. Otherwise, find out what COM port appears once driver installation has finished.) For more information on how to install the sensor software on a computer and begin using it, see the Quick Start guide.

4.2. Protocol Packet Format

4.2.1 Binary Packet Format

The binary packet size can be three or more bytes long, depending upon the nature of the command being sent to the controller. Each packet consists of an initial **"start of packet"** byte, followed by a **"command value"** specifier byte, followed by zero or more "**command data"** bytes, and terminated by a packet "**checksum value**" byte.

Each binary packet is at least 3 bytes in length and is formatted as shown in figure 1

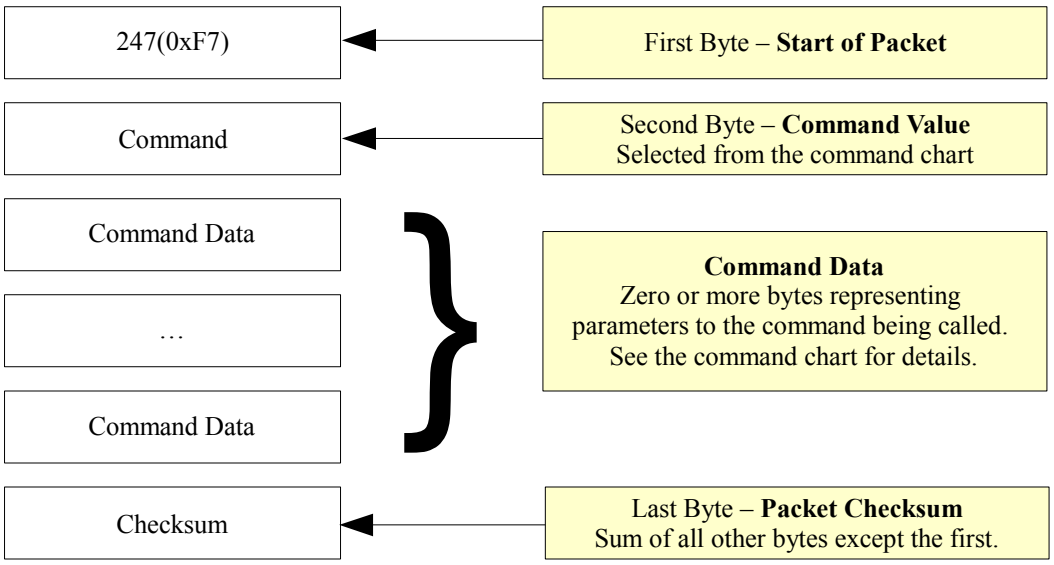


Figure 1 - Typical Binary Command Packet Format

Binary Return Values:

When a 3 Space Sensor command is called in binary mode, any data it returns will also be in binary format. For example, if a floating point number is returned, it will be returned as its 4 byte binary representation.

For information on the floating point format, go here: http://en.wikipedia.org/wiki/Single_precision_floating-point_format

Also keep in mind that integer and floating point values coming from the sensor are stored in big-endian format.

The Checksum Value:

The checksum is computed as an arithmetic summation of all of the characters in the packet (except the first byte and the checksum value itself) modulus 256. This gives a resulting checksum in the range 0 to 255. The checksum for binary packets is transmitted as a single 8-bit byte value.

4.2.2 ASCII Text Packet Format

ASCII text command packets are similar to binary command packets, but are received as a single formatted line of text. Each text line consists of the following: an ASCII colon character followed by an integral command id in decimal, followed by a list of ASCII encoded floating-point command values, followed by a terminating newline character. The command id and command values are given in decimal. The ASCII encoded command values must be separated by an ASCII comma character or an ASCII space character. Thus, legal command characters are: the colon, the comma, the period, the digits 0 through 9, the minus sign, the new-line, the space, and the backspace. When a command calls for an integer or byte sized parameter, the floating point number given for that parameter will be interpreted as being the appropriate data type. For simplicity, the ASCII encoded commands follow the same format as the binary encoded commands, but ASCII text encodings of values are used rather than raw binary encodings.

Each ASCII packet is formatted as shown in figure 2.

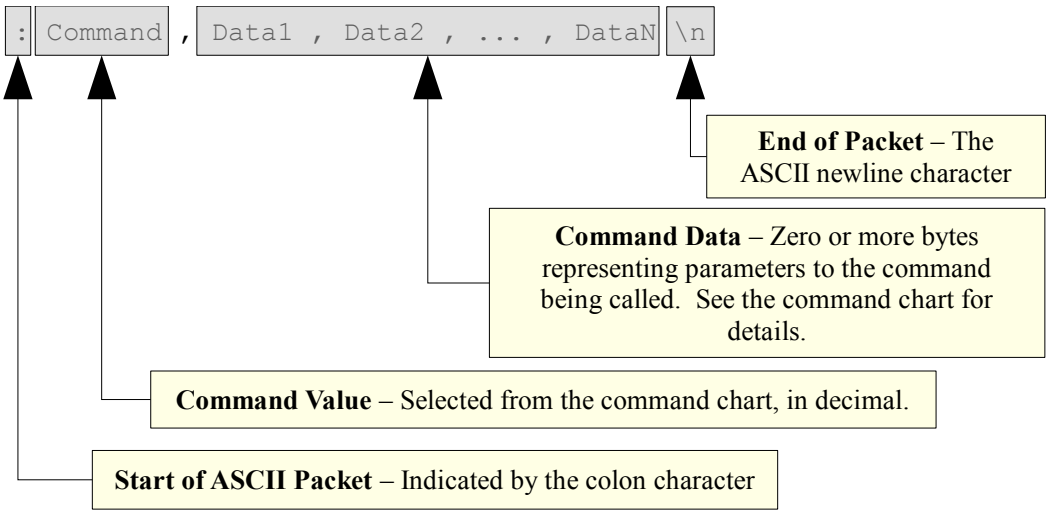

: Command , Data1 , Data2 , ... , DataN \n
End of Packet – The ASCII newline character
Command Data – Zero or more bytes representing parameters to the command being called. See the command chart for details.
Command Value – Selected from the command chart, in decimal.
Start of ASCII Packet – Indicated by the colon character

Figure 2 - Typical ASCII Command Packet Format

Thus the ASCII packet consists of the the following characters:

- **:** – the ASCII colon character signifies the start of an ASCII text packet.
- **,** – the ASCII comma character acts as a value delimiter when multiple values are specified.
- **.** – the ASCII period character is used in floating point numbers.
- **0~9** – the ASCII digits are used to in integer and floating point values.
- **-** the ASCII minus sign is used to indicate a negative number
- **\n** – the ASCII newline character is used to signify the end of an ASCII command packet.
- **\b** – the ASCII backspace character can be used to backup through the partially completed line to correct errors.

If a command is given in ASCII mode but does not have the right number of parameters, the entire command will be ignored.

Sample ASCII commands:

`:0\n`	Read orientation as a quaternion
`:106,2\n`	Set oversample rate to 2

ASCII Return Values:

All values are returned in ASCII text format when an ASCII-format command is issued. To read the return data, simply read data from the sensor until a Windows newline(a carriage return and a line feed) is encountered..

4.3. Command Chart

There are over 90 different command messages that are grouped numerically by function. Unused command message bytes are reserved for future expansion. All commands are listed in the command chart which follows.

When looking at the command charts, note the following:

- The “Data Len” field indicates the number of additional data-bytes the command expects to follow the command-byte itself. This number doesn't include the Start of Packet, Command, or Checksum bytes. Thus, the total message size can be calculated by adding three bytes to the “Data Len” listed in the table.
- Likewise, the “Return Data Len” field indicates the number of data-bytes the command delivers back to the sender once the command has finished executing.
- Under “Return Data Details”, each command lists the sort of data which is being returned and next to this in parenthesis the form this data takes. For example, a quaternion is represented by 4 floating point numbers, so a command which returns a quaternion would list “Quaternion(float x4)” for its return data details.
- Command length information only applies to binary commands, as ascii commands can vary in length.
- For quaternions, data is always returned in x, y, z, w order.
- Euler angles are always returned in pitch, yaw, roll order.
- When calling commands in ASCII mode, there is no fixed byte length for the parameter data or return data, as the length depends on the ASCII encoding.

4.3.1 Filtered Data Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
0(0x00)	Read gyroscope fusion result	Returns the final gyroscope reading output from the fusion filter.	12	Vector(float x3)	0	
1(0x01)	Read accel fusion result	Returns the final accelerometer reading output from the fusion filter.	12	Vector(float x3)	0	
2(0x02)	Read compass fusion result	Returns the final compass reading output from the fusion filter.	12	Vector(float x3)	0	
3(0x03)	Read orient fusion result	Returns the final orientation output from the fusion filter, relative to the tare orientation.	16	Quaternion(float x4)	0	
4(0x04)	Read all component fusion results	Returns the final gyroscope, accelerometer, and compass readings from the fusion filters.	36	Gyro vector(float x3), Accel vector(float x3), Compass vector(float x3)	0	
14(0x0E)	Set filter modes	Set which types of filters are enabled or disabled.	0		1	Mode bitfield(U8, bit 5 on for delta theta off, bit 4 on for orient off, bit 3 on for compass off, bit 2 on for accel off, bit 1 on for gyro off)
15(0x0F)	Read filter modes	Read which types of filters are enabled or disabled.	1	Mode bitfield(U8, bit 5 on for delta theta off, bit 4 on for orient off, bit 3 on for compass off, bit 2 on for accel off, bit 1 on for gyro off)	0	
16(0x10)	Read delta theta for last time frame(gyroscope)	Read the difference in orientation, based on the gyroscope, since the last calculation frame, as a quaternion.	16	Quaternion(float x4)	0	
17(0x11)	Read delta theta since last read(gyroscope)	Read the difference in orientation, based on the gyroscope, since the last call to this command, as a quaternion.	16	Quaternion(float x4)	0	
18(0x12)	Read delta theta for last time frame(orientation)	Read the difference in orientation, based on the fused orientation, since the last calculation frame, as a quaternion.	16	Quaternion(float x4)	0	
19(0x13)	Read delta theta since last read(orientation)	Read the difference in orientation, based on the fused orientation, since the last call to this command, as a quaternion.	16	Quaternion(float x4)	0	
20(0x14)	Read accumulated axis rotations since last frame	Read the time integrated sum of the gyroscope readings since the last calculation frame. Useful in cases where a stable rotation about a single axis is being measured, but the overall orientation does not matter.	12	Vector(float x3)	0	
21(0x15)	Read accumulated axis rotations since last read	Read the time integrated sum of the gyroscope readings since the last call to this command. Useful in cases where a stable rotation about a single axis is being measured, but the overall orientation does not matter.	12	Vector(float x3)	0	
22(0x16)	Read delta velocity for last time frame	Read the estimated change in velocity since the last calculation frame as calculated from an orientation compensated accelerometer reading.	12	Vector(float x3)	0	
23(0x17)	Read delta velocity since last read	Read the estimated change in velocity since the last call to this command as calculated from an orientation compensated accelerometer reading.	12	Vector(float x3)	0	
24(0x18)	Read orientation filtered accelerometer vector	Read the orientation filtered accelerometer vector. This should point in roughly the same direction as the accelerometer vector, but it is much more stable as it comes from the fused orientation.	12	Vector(float x3)	0	
25(0x19)	Read orientation filtered compass vector	Read the orientation filtered compass vector. This should point in roughly the same direction as the accelerometer vector, but it is much more stable as it comes from the fused orientation.	12	Vector(float x3)	0	

4.3.2 Raw Data Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
32(0x20)	Read single raw gyro	Read the raw value from a single gyroscope.	6	Raw vector(short x3)	1	Index(U8)
33(0x21)	Read single accel raw	Read the raw value from a single accelerometer.	6	Raw vector(short x3)	1	Index(U8)
34(0x22)	Read single compass raw	Read the raw value from a single compass.	6	Raw vector(short x3)	1	Index(U8)
35(0x23)	Read all gyros raw	Read the raw values from all gyroscopes.	96	Raw vector(short x3) x16	0	
36(0x24)	Read all accels raw	Read the raw values from all accelerometer.	24	Raw vector(short x3) x4	0	
37(0x25)	Read all compasses raw	Read the raw values from all compasses.	24	Raw vector(short x3) x4	0	
38(0x26)	Read all raw with timestamp	Read the raw values from all sensors,along with a timestamp.	148	Timestamp(U32),Raw vector(short x3) x24	0	
39(0x27)	Read raw barometer	Read the raw barometer value.	2	Raw value(short)	0	

4.3.3 Calibrated Data Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
48(0x30)	Read single gyro	Read the calibrated value from a single gyroscope.	12	Vector(float x3)	1	Index(U8)
49(0x31)	Read single accel	Read the calibrated value from a single accelerometer.	12	Vector(float x3)	1	Index(U8)
50(0x32)	Read single compass	Read the calibrated value from a single compass.	12	Vector(float x3)	1	Index(U8)
51(0x33)	Read all gyros	Read the calibrated values from all gyroscopes.	192	Vector(float x3) x16	0	
52(0x34)	Read all accels	Read the calibrated values from all accelerometer.	48	Vector(float x3) x4	0	
53(0x35)	Read all compasses	Read the calibrated values from all compasses.	48	Vector(float x3) x4	0	
54(0x36)	Read all with timestamp	Read the calibrated values from all sensors,along with a timestamp.	292	Timestamp(U32),Vector(float x3) x24	0	
56(0x38)	Read gyro temperature C	Read the calibrated temperature of a gyroscope in degrees C.	4	Temperature(float)	1	Index(U8)
57(0x39)	Read gyro temperature F	Read the calibrated temperature of a gyroscope in degrees F.	4	Temperature(float)	1	Index(U8)
58(0x3A)	Read all gyro temperatures C	Read the calibrated temperatures of all gyroscopes in degrees C.	64	Temperature(float)*16	0	

4.3.4 Orientation Fusion Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
64(0x40)	Tare with current orientation	Set the tare orientation to the current orientation.	0		0	
65(0x41)	Tare with quaternion	Set the tare orientation to the give quaternion.	0		16	Quaternion(float x4)
66(0x42)	Read tare orientation	Read the tare orientation.	0	Quaternion(float x4)	0	
67(0x43)	Read covariance matrix	Read the current Kalman filter covariance matrix.	0	Covariance matrix(float x16)	0	
68(0x44)	Set gyro enabled	Enable or disable a gyroscope.	0		2	Index(U8), State(U8, 0 for disabled, 1 for enabled)
69(0x45)	Read gyro enabled	Read the enabled state of a gyroscope.	1	State(U8, 0 for disabled, 1 for enabled)	1	Index(U8)
70(0x46)	Set accel enabled	Enable or disable a accelerometer.	0		2	Index(U8), State(U8, 0 for disabled, 1 for enabled)
71(0x47)	Read accel enabled	Read the enabled state of a acclerometer.	1	State(U8, 0 for disabled, 1 for enabled)	1	Index(U8)
72(0x48)	Set compass enabled	Enable or disable a compass.	0		2	Index(U8), State(U8, 0 for disabled, 1 for enabled)
73(0x49)	Read compass enabled	Read the enabled state of a compass.	1	State(U8, 0 for disabled, 1 for enabled)	1	Index(U8)
74(0x4A)	Set reference vectors	Set accelerometer and compass reference vectors.	0		24	Accel ref vector(float x3), Compass ref vector(float x3)
75(0x4B)	Read reference vectors	Read accelerometer and compass reference vectors.	24	Accel ref vector(float x3), Compass ref vector(float x3)	0	

4.3.5 Gyro Fusion Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
96(0x60)	Set gyro fusion method	Set the filtering method the gyroscope fusion uses.	0		1	Method(U8, 0 for average, 3 for filtered)
97(0x61)	Read gyro fusion method	Read the filtering method the gyroscope fusion uses.	1	Method(U8, 0 for average, 3 for filtered)	0	
98(0x62)	Set deadband	Enable or disable the deadband.	0		1	State(U8, 0 for disabled, 1 for enabled)
99(0x63)	Read deadband	Read the enabled state of the deadband.	1	State(U8, 0 for disabled, 1 for enabled)	0	
100(0x64)	Set deadband threshold	Set the threshold used for stillness by the deadband.			4	Stillness(float)
101(0x65)	Read deadband threshold	Read the threshold used for stillness by the deadband.	4	Stillness(float)		
102(0x66)	Set bias tracking mode	Set the bias tracking method used by gyroscope fusion.	0		1	Method(U8, 0 for none, 1 for gyro bias only, 2 for fusion bias only, 3 for both, 7 for both with no temp bias tracking)
103(0x67)	Read bias tracking mode	Read the bias tracking method used by gyroscope fusion.	1	Method(U8, 0 for none, 1 for gyro bias only, 2 for fusion bias only, 3 for both, 7 for both with no temp bias tracking)	0	
104(0x68)	Set fusion bias	Set the fused gyroscope reading bias.	0		12	Vector(float x3)
105(0x69)	Read fusion bias	Read the fused gyroscope reading bias.	12	Vector(float x3)	0	
106(0x6A)	Read sensor stillness	Read the stillness value used by bias tracking and the deadband.	4	Stillness(float)	0	
107(0x6B)	Set bias tracking stillness threshold	Set the threshold used for stillness by bias tracking.	0		4	Stillness(float)
108(0x6C)	Read bias tracking stillness threshold	Read the stillness threshold used by bias tracking.	4	Stillness(float)	0	

4.3.6 Calibration Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
128(0x80)	Calibrate all gyro biases	Run gyroscope bias calibration. Both LEDs will turn purple during this process. Leave the sensor as still as possible until the LEDs revert to their normal colors.	0		0	
130(0x82)	Set gyro calibration parameters	Set bias and scale parameters for a gyroscope.	0		49	Index(U8), Bias(float x3), Matrix(float x9)
131(0x83)	Read gyro calibration parameters	Read bias and scale parameters for a gyroscope.	48	Bias(float x3), Matrix(float x9)	1	Index(U8)
132(0x84)	Set gyro temperature parameters	Set temperature parameters for a gyroscope.	0		49	Index(U8), Bias per degree(float x3), Linear scale per degree(float x9)
133(0x85)	Read gyro temperature parameters	Read temperature parameters for a gyroscope.	48	Bias per degree(float x3), Linear scale per degree(float x9)	1	Index(U8)
134(0x86)	Set accel calibration parameters	Set bias and scale parameters for an accelerometer.	0		49	Index(U8), Bias(float x3), Matrix(float x9)
135(0x87)	Read accel calibration parameters	Read bias and scale parameters for an accelerometer.	48	Bias(float x3), Matrix(float x9)	1	Index(U8)
136(0x88)	Set accel temperature parameters	Set temperature parameters for an accelerometer.	0		49	Index(U8), Bias per degree(float x3), Linear scale per degree(float x9)
137(0x89)	Read accel temperature parameters	Read temperature parameters for an accelerometer.	48	Bias per degree(float x3), Linear scale per degree(float x9)	1	Index(U8)
138(0x8A)	Set compass calibration parameters	Set bias and scale parameters for a compass.	0		49	Index(U8), Bias(float x3), Matrix(float x9)
139(0x8B)	Read compass calibration parameters	Read bias and scale parameters for a compass.	48	Bias(float x3), Matrix(float x9)	1	Index(U8)
140(0x8C)	Set compass temperature parameters	Set temperature parameters for a compass.	0		49	Index(U8), Bias per degree(float x3), Linear scale per degree(float x9)
141(0x8D)	Read compass temperature parameters	Read temperature parameters for a compass.	48	Bias per degree(float x3), Linear scale per degree(float x9)	1	Index(U8)
142(0x8E)	Set calibration mode	Set calibration mode for sensors.	0		1	Mode(U8, 0 for scale and bias only, 1 for vector correction)
143(0x8F)	Read calibration mode	Read calibration mode for sensors.	1	Mode(U8, 0 for scale and bias only, 1 for vector correction)	0	
144(0x90)	Set lookup table vertex	Set a vertex in the vector lookup table.	0		15	Value type(U8, 0 for compass, 1 for accelerometer), Index(short), Value(float x3)
145(0x91)	Read lookup table vertex	Read a vertex in the vector lookup table.	12	Value(float x3)	3	Value type(U8, 0 for compass, 1 for accelerometer), Index(short)
146(0x92)	Set ortho calibration from current orient	Set an ortho calibration point using the current orientation.	0		0	
148(0x94)	Set ortho calibration point	Set an ortho calibration point manually.	0		14	Type(U8, 0 for compass, 1 for accel), Index(U8), Vector(float x3)
149(0x95)	Read ortho calibration point	Read an ortho calibration point.	12	Vector(float x3)	2	Type(U8, 0 for compass, 1 for accel), Index(U8)
150(0x96)	Perform ortho calibration	Use the ortho calibration data to fill the lookup table.	0		0	
151(0x97)	Clear ortho calibration	Clear the ortho calibration data.	0		0	
152(0x98)	Set gyro range	Set the range of a single gyroscope.	0		2	Index(U8), Range(U8, 0 for 250 dps, 1 for 500 dps, 2 for 2000 dps)
153(0x99)	Read gyro range	Read the range of a single gyroscope.	1	Range(U8, 0 for 250 dps, 1 for 500 dps, 2 for 2000 dps)	1	Index(U8)
154(0x9A)	Set accel range	Set the range of a single accelerometer.	0		2	Index(U8), Range(U8, 0 for 2G, 1 for 4G, 2 for 8G)
155(0x9B)	Read accel range	Read the range of a single accelerometer.	1	Range(U8, 0 for 2G, 1 for 4G, 2 for 8G)	1	Index(U8)
156(0x9C)	Set compass range	Set the range of a single compass.	0		2	Index(U8), Range(U8, 0 for 10 G)
157(0x9D)	Read compass range	Read the range of a single compass.	1	Range(U8, 0 for 10 G)	1	Index(U8)
158(0x9E)	Set gyro oversample rate	Set the current oversample rate for all gyroscopes.	0		1	Oversample rate(U8, 1 to 10, 1 being no oversampling)
159(0x9F)	Read gyro oversample rate	Read the current oversample rate for all gyroscopes.	1	Oversample rate(U8, 1 to 10, 1 being no oversampling)	0	

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
160(0xa0)	Set temperature filter alpha	Sets the alpha value of the temperature's low pass filter.	0		4	Alpha value(float)
161(0xa1)	Read temperature filter alpha	Reads the alpha value of the temperature's low pass filter.	4	Alpha value(float)	0	

4.3.7 System Commands

Command	Description	Long Description	Return Data Len	Return Data Details	Data Len	Data Details
224(0xe0)	Restore factory settings	Restore all settings to their defaults. Does not automatically commit this.	0		0	
225(0xe1)	Commit settings	Commit current changes to flash.	0		0	
226(0xe2)	Software reset	Reset the sensor from software.	0		0	
227(0xe3)	Set idle mode	Pauses or resumes the execution of all filters.	0		1	Mode(0 for no idle, 1 for idle)
228(0xe4)	Read idle mode	Reads the state of execution of the filters.	1	Mode(0 for no idle, 1 for idle)	0	
229(0xe5)	Enter bootloader mode	Put the sensor into bootloader mode.	0		0	
230(0xe6)	Read hardware id	Read hardware version information.	32	Version string	0	
231(0xe7)	Read software id	Read software version information.	32	Version string	0	
232(0xe8)	Set serial rate	Set the communication rate of a serial port.	0		5	Index(U8, 1 for RS422, 0 for debug port), Baud rate(U32)
233(0xe9)	Read serial rate	Read the communication rate of a serial port.	4	Baud rate(U32)	1	Index(U8, 1 for RS422, 0 for debug port)
235(0xeb)	Read current update rate	Read the current time taken by the filter loop per cycle, in microseconds.	4	Update rate(U32, microseconds)	0	
236(0xec)	Set serial number	Set the serial number.	0		8	Code(Byte x4), Serial number(U32)
237(0xed)	Read serial number	Read the serial number.	4	Serial number(U32)	0	
238(0xee)	Set LED color	Set the color of an LED.	0		4	LED index(U8, 0 or 1), Red(U8), Green(U8), Blue(U8)
239(0xef)	Read LED color	Read the color of an LED.	4	Red(U8), Green(U8), Blue(U8)	1	LED index(U8, 0 or 1)

Appendix

USB Connector

The 3-Space Sensor has a 5-pin USB Type-B jack and can be connected via a standard 5-pin mini USB cable.

RS422 Connector

The RS422 connector provides a means to provide to and communicate with the 3-Space Sensor via full duplex RS422 signals. The RS422 connector is a 6-pin 2mm Hirose part number DF3A-6P-2DS and allows connection of several style mating connector including IDC and crimp pin options. See the DF3A series datasheet for all mating connector options.

The signals of the RS422 connector are assigned as follows:

Signal Number	Signal Description
1	+3.3 ~ +6 vdc (External Power Input)
2	RS422 A Signal (RxD+ into unit)
3	RS422 B Signal (RxD- into unit)
4	RS422 Z Signal (TxD- from unit)
5	RS422 Y Signal (TxD+ from unit)
6	Gnd (Shared power and signal ground)

Mating plugs are available from Yost Engineering, Inc. or from other electronics vendors.

Note that the RS422 power input is provided as a convenient way to provide power along with communications via a single connector. Thus, the external power input is only required when the unit is not being powered via USB.

Hex / Decimal Conversion Chart

		Second Hexadecimal digit															
		0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***A***	***B***	***C***	***D***	***E***	***F***
First Hexadecimal Digit	***0***	000	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015
	1	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
	2	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047
	3	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063
	4	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079
	5	080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095
	6	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111
	7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
	8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
	B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
	C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
	D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
	E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
	F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Notes:

Serial Number: _____

YEI Technology

630 Second Street
Portsmouth, Ohio 45662

Toll-Free: 888-395-9029
Phone: 740-355-9029

www.YeiTechnology.com
www.3SpaceSensor.com

Patents Pending
©2007-2012 Yost Engineering, Inc.
Printed in USA